Practical Malware Analysis & Triage

Malware Analysis Report

WannaCry Ransomware Cryptoworm

December 2021 | An00bRektn | v1.1

Table of Contents

Executive Summary

SHA256 hash	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

WannaCry is a ransomware sample first identified on May 12th, 2017. It is a C++-compiled cryptoworm that runs on the x64 and x86 Windows operating systems. The sample consists of a main payload that unpacks an additional payload, along with helper files, to encrypt files on a host and demand a ransom for decryption. WannaCry propagates across a network by leveraging a known vulnerability in the Windows operating system to give full Administrator access, that has since been patched by Microsoft.

Symptoms of infection include, but are not limited to:

- Files with the '`.WNRY`' extension and a visibly different wallpaper saying that files have been encrypted
- an executable named "`@WanaDecryptor@.exe`" asking for payment to a given Bitcoin wallet
- A new, hidden directory with a name of random characters located in `C:\ProgramData\`
- Creation of a new service with the same name as the hidden folder in `C:\ProgramData\` used for persistence
- Shadow copies of the file system removed from disk
- A `taskhsvc.exe` executable listening on port 9050

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

High-Level Technical Summary

WannaCry consists of a main payload that attempts to contact its callback URL (hxxp://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com). If the call back is **unsuccessful**, it unpacks its next stage and other "helper" files into `C:\ProgramData\[RANDOM_STRING]`. The flow of execution is shown in the diagram below.

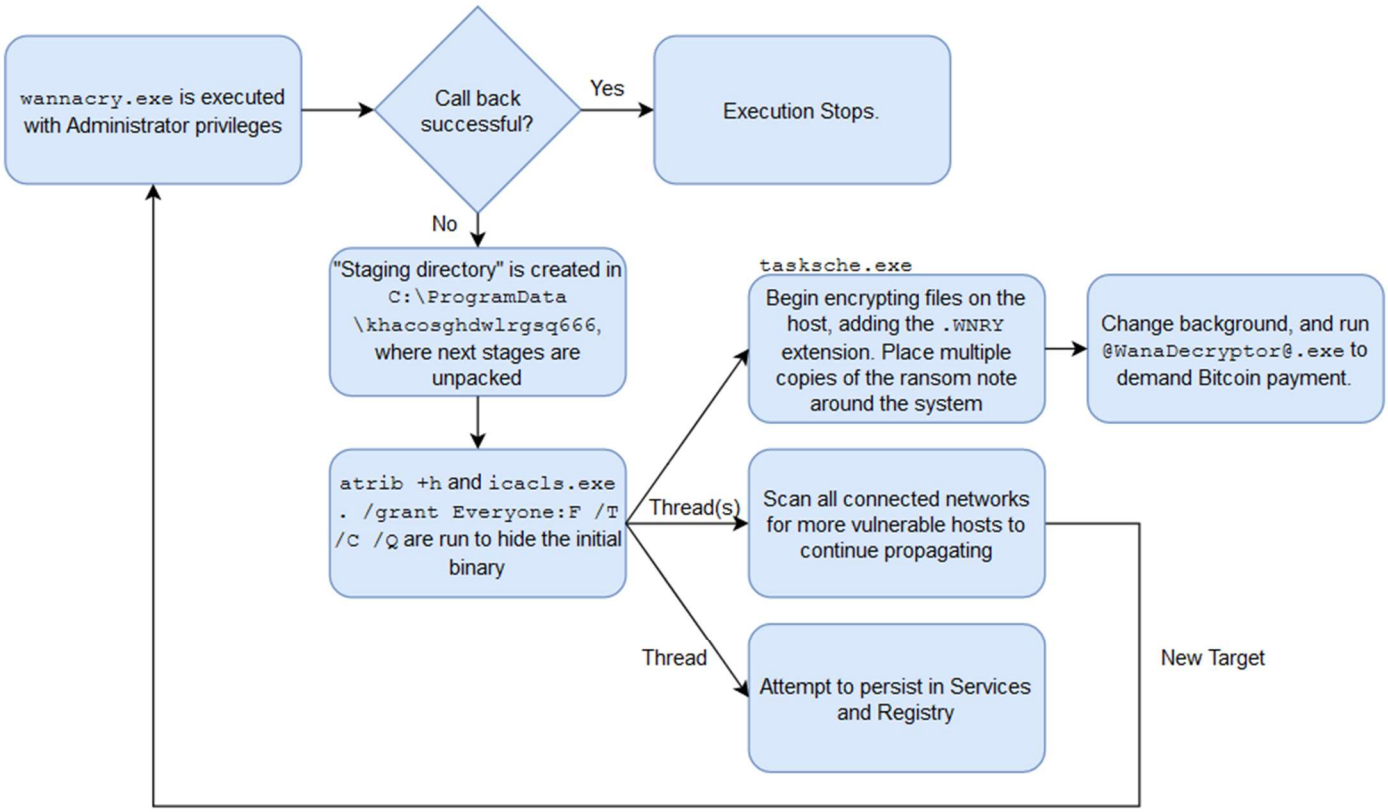


Fig 1: Flowchart depicting the execution of WannaCry.

WannaCry leverages MS17-010 (dubbed "EternalBlue") to propagate across the network, which abuses a vulnerability in the Windows SMBv1 protocol. A patch from Microsoft had already been released prior to the initial discovery of WannaCry, however, hosts that have not applied the vendor patch are susceptible to the sample.

Malware Composition

WannaCry consists of the following stages:

File Name	SHA256 Hash
wannacry.exe	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
tasksche.exe	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

wannacry.exe

The initial executable that runs after a successful EternalBlue exploit, and failure to reach the callback URL. This executable is primarily responsible for the propagation as opposed to the encryption. After creating the staging directory and attempting to persist, `wannacry.exe` will proceed to search for other hosts on all connected networks using multiple threads of execution.

tasksche.exe

The second stage executable that runs after it has been unpacked from wannacry.exe. This binary conducts most of the malicious operations on the host, primarily the encryption of the files on the compromised host.

Additional Files

As mentioned earlier, a staging directory is created in C:\ProgramData once the call to the "kill switch" URL has failed. This directory contains a number of files outside of the two stages that are not explicitly additional stages for the malware.

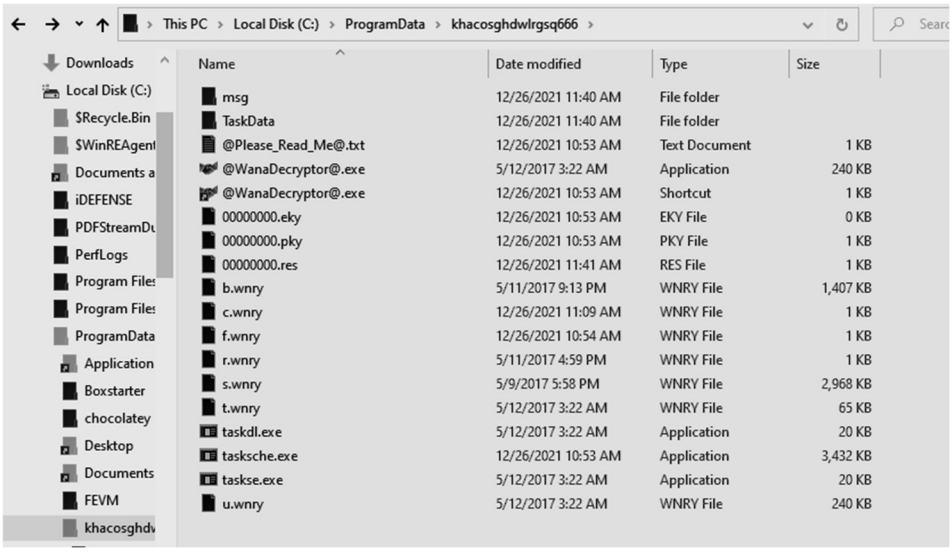


Fig 2: Screenshot from the system explorer of the files in the "staging" directory.

Listed below are the files typically found in this directory and their function.

- `msg/`: The directory containing various translations of the ransom note
- `TaskData/`: Contains various files used in the Tor browser program
- `00000000.[eky | pky | res]`: Purpose unknown; Seem to be various files with keys used in the encryption process
- `b.wnry`: The desktop background image used
- `c.wnry`: Contains data for certain “dark web” sites, a bitcoin wallet address, and a link to a Tor browser installation
- `r.wnry`: The README file to instruct victims how to decrypt their files
- `s.wnry`: Contains data with references to the Tor program
- `t.wnry`: Contains
- `u.wnry`: The Decryptor executable that shows the ransom note
- `taskdl.exe`: Locates files that have already been encrypted
- `tasksche.exe`: The second stage for the malware
- `taskse.exe`: Purpose unknown; seems to run the executable that gets passed in as an argument

Basic Static Analysis

Preliminary Information Gathering

Architecture

```
C:\Users\sreisz\Desktop
λ file Ransomware.wannacry.exe.malz
Ransomware.wannacry.exe.malz: PE32 executable (GUI) Intel 80386, for MS Windows
```

Hashes

Type	Hash
MD5	db349b97c37d22f5ea1d1841e3c89eb4
SHA256	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

Strings

Below are annotated screenshots of interesting strings found in the sample.

```
__USERID__PLACEHOLDER__@
h6agLCqPqVyXi2VSQ8O6Yb9ijBX54jY6KM+sz33NmS6TK8XlOk920s0E0aajOV++wrR92ds1FOLBO+evLPj4
sIvAjLvaLdgk8+BlNZs8PMa9bQ340J83nx1p4f+GLpbxUy...
[trim]...vXbwtmRkgiGneql4mBYmKDzcXCkp/tjnL6/KriY81gMHN4G9ulMunxVyF8wybDcifTOxtarjLXV
RuC1Y7vzYaEuHT
SMB3
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
h54WfF9cGigWFEx92bzmOd0UOaZlMDdU2F4F2+6qn9/ZDSqJksnLIfbdOiMA3D+1qUTSrerHhgCcS2PibZuz
q9y+eWLOzmwXaWqkEMg2LUA3HWJN4+Sf5DkSGjBmXQb0UQ...
[trim]...czk25ArAZcQXDX1MRxY20HuT3rhmYYLpiuJX/mu7wb6CGWZ4i6/eolXB3sb3ucvGEzAheJm9zxn
H3/tcqpC4MtJe/6OAawtD+e362d6
bbCUB+5x4jIXypy61OlDcDWgbfIXcwcI02u15qZXg4cV/VjsDiEQARjmMebJBucJxC7HA9GSmUefyzAun9fL
ULv3RbywhnNACbSX9hbRj/rxlAlfKv1cBRDwhcdL9p+vmw...
[trim]...xLi7tEokjkrjiuxTJ7VOLMMoSqihIRgpTXkEvW4yy3O1fgQ+bAb0PNcCPaSxznfpGq9Rcq8uTkC
gqDKEBujpjKKYi4BHd
```

Interesting, long, encrypted(?) strings
Repeats multiple times in the output

```
\\%s\IPC$
Microsoft Base Cryptographic Provider v1.0
%d.%d.%d.%d
```

Potential calls to other SMB
Shares on local network

```
mssecsvc2.0
Microsoft Security Center (2.0) Service
%s -m security
```

Further supporting service persistence
hypothesis

```
C:\%s\qeriuwjhrf
C:\%s\%s
```

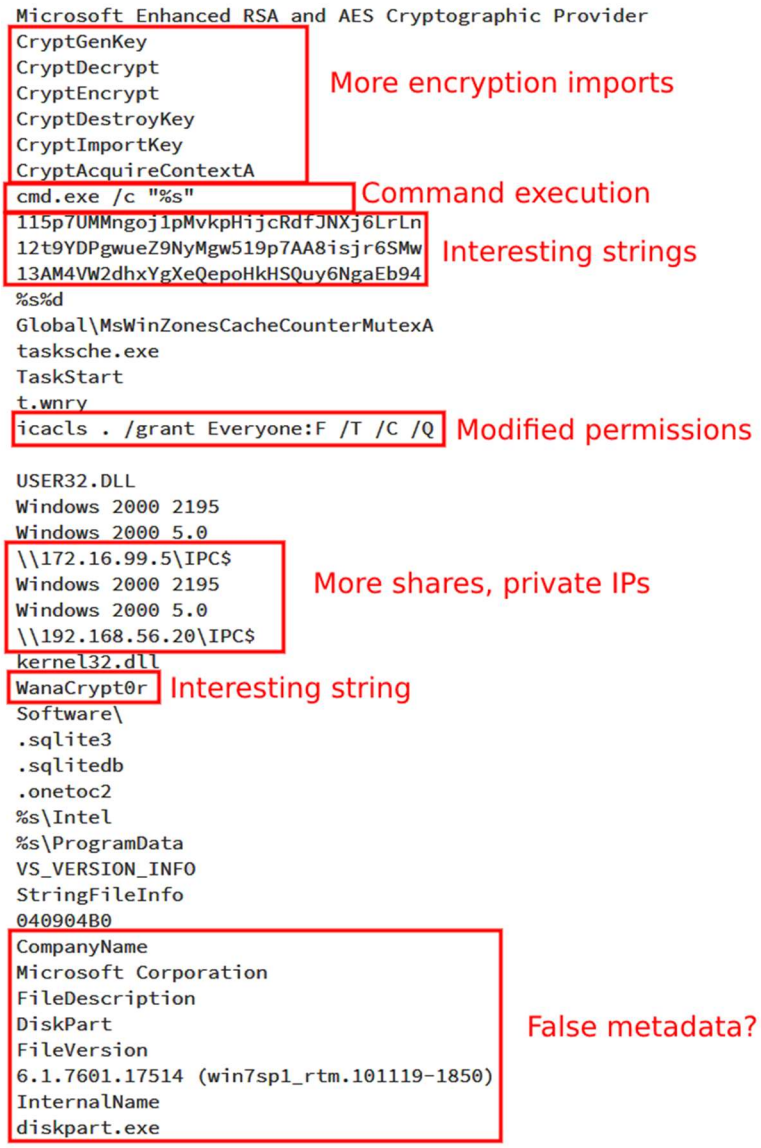
Interesting path

```
WINDOWS
tasksche.exe
CloseHandle
WriteFile
CreateFileA
CreateProcessA
```

Not the actual Windows Task Scheduler, potential second stage?

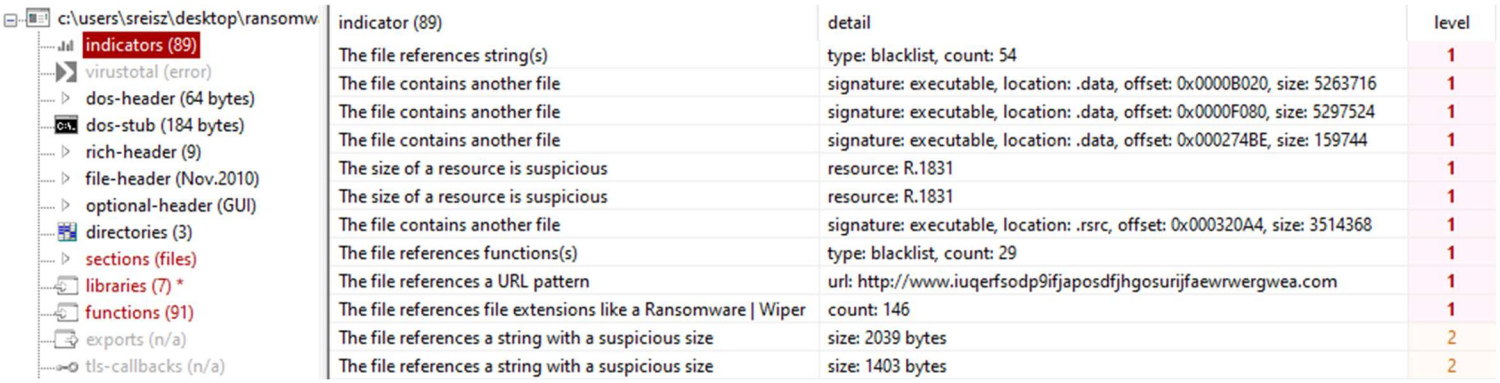
```
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
```

Potential callback URL


Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
More encryption imports
cmd.exe /c "%s"
Command execution
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Interesting strings
%s%d
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
t.wnry
icacls . /grant Everyone:F /T /C /Q
Modified permissions
USER32.DLL
Windows 2000 2195
Windows 2000 5.0
\\172.16.99.5\IPC$
Windows 2000 2195
Windows 2000 5.0
\\192.168.56.20\IPC$
More shares, private IPs
kernel32.dll
WanaCrypt0r
Interesting string
Software\
.sqlite3
.sqlitedb
.onetoc2
%s\Intel
%s\ProgramData
VS_VERSION_INFO
StringFileInfo
040904B0
CompanyName
Microsoft Corporation
FileDescription
DiskPart
FileVersion
6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName
diskpart.exe
False metadata?

PEStudio

PEStudio Indicators



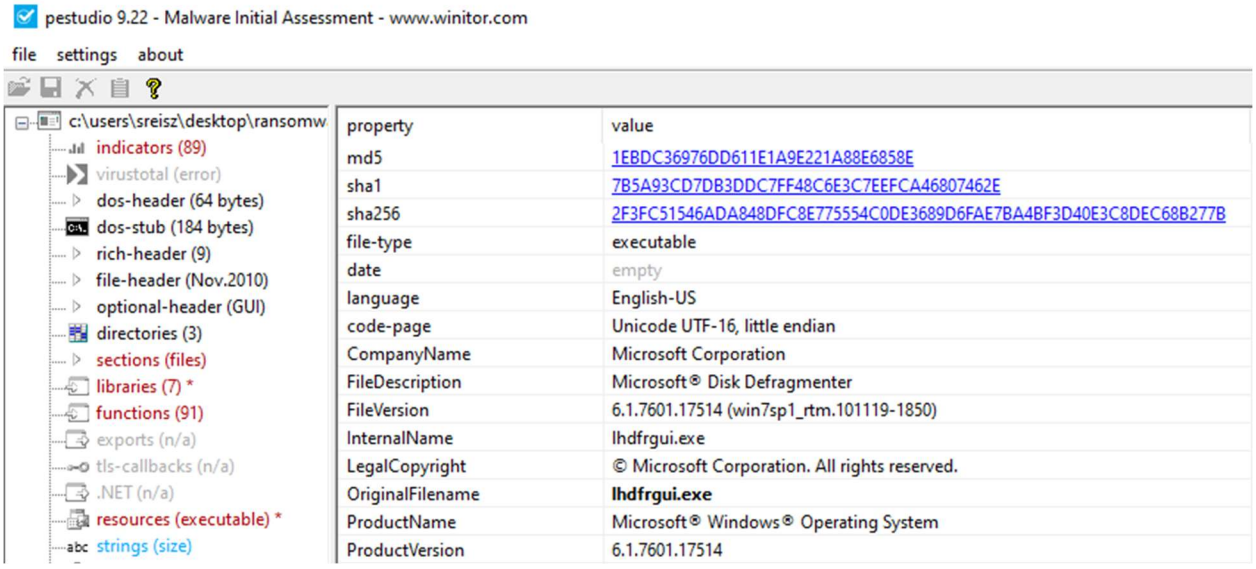
c:\users\sreisz\desktop\ransomw
- indicators (89)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (184 bytes)
- rich-header (9)
- file-header (Nov.2010)
- optional-header (GUI)
- directories (3)
- sections (files)
- libraries (7) *
- functions (91)
- exports (n/a)
- tls-callbacks (n/a)

indicator (89)	detail	level
The file references string(s)	type: blacklist, count: 54	1
The file contains another file	signature: executable, location: .data, offset: 0x0000B020, size: 5263716	1
The file contains another file	signature: executable, location: .data, offset: 0x0000F080, size: 5297524	1
The file contains another file	signature: executable, location: .data, offset: 0x000274BE, size: 159744	1
The size of a resource is suspicious	resource: R.1831	1
The size of a resource is suspicious	resource: R.1831	1
The file contains another file	signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368	1
The file references functions(s)	type: blacklist, count: 29	1
The file references a URL pattern	url: http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com	1
The file references file extensions like a Ransomware | Wiper	count: 146	1
The file references a string with a suspicious size	size: 2039 bytes	2
The file references a string with a suspicious size	size: 1403 bytes	2

Fig. 3.1

- Suggests that there might be more stages to be unpacked
- “The file references file extensions like a Ransomware | Wiper” correlates with this being ransomware
- “The file references a string with a suspicious size” falls in line with some of the long, encrypted strings found earlier

Version



pestudio 9.22 - Malware Initial Assessment - www.winitor.com

file settings about

c:\users\sreisz\desktop\ransomw
- indicators (89)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (184 bytes)
- rich-header (9)
- file-header (Nov.2010)
- optional-header (GUI)
- directories (3)
- sections (files)
- libraries (7) *
- functions (91)
- exports (n/a)
- tls-callbacks (n/a)
- .NET (n/a)
- resources (executable) *
- strings (size)

property	value
md5	1EBDC36976DD611E1A9E221A88E6858E
sha1	7B5A93CD7DB3DDC7FF48C6E3C7EEFCA46807462E
sha256	2F3FC51546ADA848DFC8E775554C0DE3689D6FAE7BA4BF3D40E3C8DEC68B277B
file-type	executable
date	empty
language	English-US
code-page	Unicode UTF-16, little endian
CompanyName	Microsoft Corporation
FileDescription	Microsoft® Disk Defragmenter
FileVersion	6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName	lhdfrgui.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	**lhdfrgui.exe**
ProductName	Microsoft® Windows® Operating System
ProductVersion	6.1.7601.17514

Fig. 3.2

The executable seems to have metadata to masquerade as an official Microsoft product.

Imports

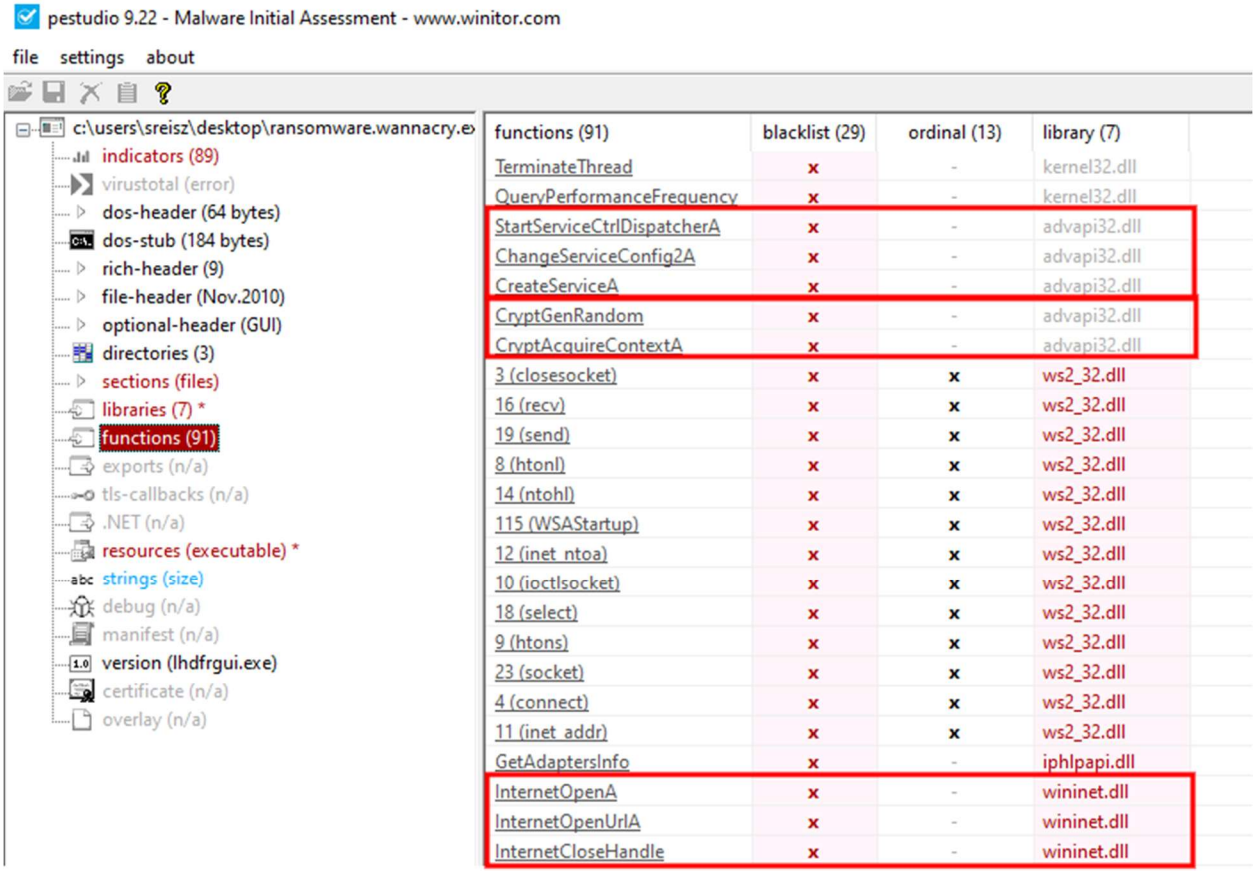


Fig 3.3

- The calls related to Services potentially indicate some kind of service-based persistence
- `CryptGenRandom` and `CryptAcquireContextA` fall in line with this sample being ransomware. Additional calls to the cryptographic functions in the Windows API were also found
- `InternetOpenA` and other imports follow the typical setup for a malicious binary to access some web resource.

Basic Dynamic Analysis

Initial Detonation – With Internet Simulation

After running the sample with Administrator permissions, the following packets were captured using Wireshark, from the connected Remnux VM.

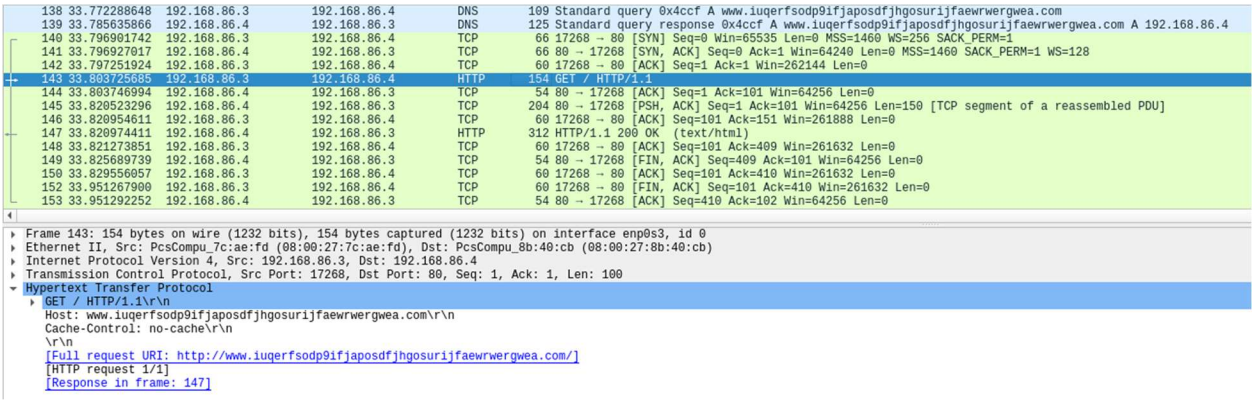


Fig 4.1: Packet capture showing the HTTP request to the callback URL.

However, monitoring on the host using `procmon` showed that the sample terminated execution after this point.

Fig 4.2: The sample's metadata claims to be an officially licensed Microsoft tool.

No additional artifacts were found on the system during initial detonation in the presence of an internet simulation.

Initial Detonation – Without Internet Simulation

After disabling the internet simulation and running the sample again, DNS requests to the same URL are observed again. Since internet simulation is disabled, these requests fail.

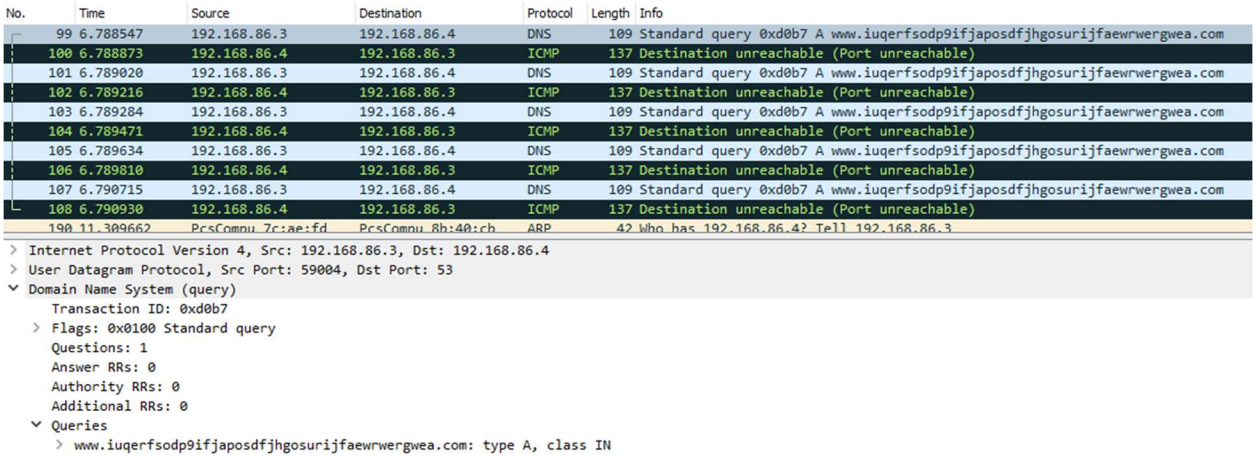
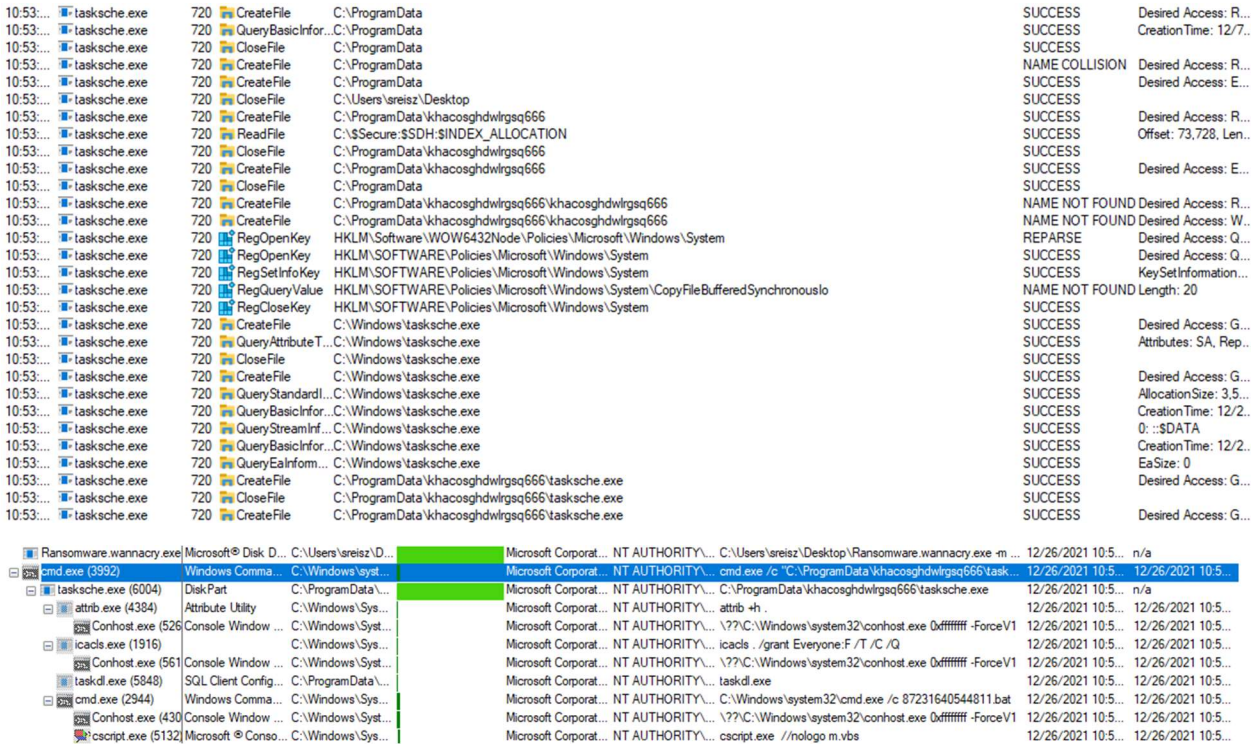


Fig 4.3: Failed requests to the callback URL.

However, the sample continues to execute. New files are observed on the Desktop.

Fig 4.4: Screenshot of the Desktop, where new files, such as @WanaDecryptor@.exe have been placed.

Using `procmon`, we note the creation of a new file, tasksche.exe, in `C:\Windows`. This file creates a new, hidden directory: `C:\ProgramData\khacosghdwlrgsq666`. The contents of this directory can be seen in Figure 2.



10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData	SUCCESS	Desired Access: R...	
10:53:...	tasksche.exe	720	QueryBasicInfor...	C:\ProgramData	SUCCESS	CreationTime: 12/7...	
10:53:...	tasksche.exe	720	CloseFile	C:\ProgramData	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData	NAME COLLISION	Desired Access: R...	
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData	SUCCESS	Desired Access: E...	
10:53:...	tasksche.exe	720	CloseFile	C:\Users\sreisz\Desktop	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666	SUCCESS	Desired Access: R...	
10:53:...	tasksche.exe	720	ReadFile	C:\$Secure:$SDH:$INDEX_ALLOCATION	SUCCESS	Offset: 73,728, Len...	
10:53:...	tasksche.exe	720	CloseFile	C:\ProgramData\khacosghdwlrgsq666	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666	SUCCESS	Desired Access: E...	
10:53:...	tasksche.exe	720	CloseFile	C:\ProgramData	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666\khacosghdwlrgsq666	NAME NOT FOUND	Desired Access: R...	
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666\khacosghdwlrgsq666	NAME NOT FOUND	Desired Access: W...	
10:53:...	tasksche.exe	720	RegOpenKey	HKLM\Software\WOW6432Node\Policies\Microsoft\Windows\System	REPARSE	Desired Access: Q...	
10:53:...	tasksche.exe	720	RegOpenKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\System	SUCCESS	Desired Access: Q...	
10:53:...	tasksche.exe	720	RegSetInfoKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\System	SUCCESS	KeySetInformation...	
10:53:...	tasksche.exe	720	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\System\CopyFileBufferedSynchronousIo	NAME NOT FOUND	Length: 20	
10:53:...	tasksche.exe	720	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\System	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\Windows\tasksche.exe	SUCCESS	Desired Access: G...	
10:53:...	tasksche.exe	720	QueryAttributeT...	C:\Windows\tasksche.exe	SUCCESS	Attributes: SA, Rep...	
10:53:...	tasksche.exe	720	CloseFile	C:\Windows\tasksche.exe	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\Windows\tasksche.exe	SUCCESS	Desired Access: G...	
10:53:...	tasksche.exe	720	QueryStandardI...	C:\Windows\tasksche.exe	SUCCESS	AllocationSize: 3,5...	
10:53:...	tasksche.exe	720	QueryBasicInfor...	C:\Windows\tasksche.exe	SUCCESS	CreationTime: 12/2...	
10:53:...	tasksche.exe	720	QueryStreamInf...	C:\Windows\tasksche.exe	SUCCESS	0: ::$DATA	
10:53:...	tasksche.exe	720	QueryBasicInfor...	C:\Windows\tasksche.exe	SUCCESS	CreationTime: 12/2...	
10:53:...	tasksche.exe	720	QueryEaInform...	C:\Windows\tasksche.exe	SUCCESS	EaSize: 0	
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666\tasksche.exe	SUCCESS	Desired Access: G...	
10:53:...	tasksche.exe	720	CloseFile	C:\ProgramData\khacosghdwlrgsq666\tasksche.exe	SUCCESS		
10:53:...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrgsq666\tasksche.exe	SUCCESS	Desired Access: G...	

Ransomware.wannacry.exe	Microsoft® Disk D...	C:\Users\sreisz\D...	Microsoft Corporat...	NT AUTHORITY\...	C:\Users\sreisz\Desktop\Ransomware.wannacry.exe -m ...	12/26/2021 10:5...	n/a
cmd.exe (3992)	Windows Comma...	C:\Windows\syst...	Microsoft Corporat...	NT AUTHORITY\...	cmd.exe /c "C:\ProgramData\khacosghdwlrgsq666\task...	12/26/2021 10:5...	12/26/2021 10:5...
tasksche.exe (6004)	DiskPart	C:\ProgramData\...	Microsoft Corporat...	NT AUTHORITY\...	C:\ProgramData\khacosghdwlrgsq666\tasksche.exe	12/26/2021 10:5...	n/a
attrib.exe (4384)	Attribute Utility	C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	attrib +h .	12/26/2021 10:5...	12/26/2021 10:5...
Conhost.exe (526	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	12/26/2021 10:5...	12/26/2021 10:5...
icacls.exe (1916)		C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	icacls . /grant Everyone:F /T /C /Q	12/26/2021 10:5...	12/26/2021 10:5...
Conhost.exe (561	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	12/26/2021 10:5...	12/26/2021 10:5...
taskdl.exe (5848)	SQL Client Config...	C:\ProgramData\...	Microsoft Corporat...	NT AUTHORITY\...	taskdl.exe	12/26/2021 10:5...	12/26/2021 10:5...
cmd.exe (2944)	Windows Comma...	C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	C:\Windows\system32\cmd.exe /c 87231640544811.bat	12/26/2021 10:5...	12/26/2021 10:5...
Conhost.exe (430	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	12/26/2021 10:5...	12/26/2021 10:5...
cscript.exe (5132	Microsoft ® Conso...	C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	cscript.exe //nologo m.vbs	12/26/2021 10:5...	12/26/2021 10:5...

Fig 4.5: Procmon logs detailing the creation of the staging directory.

Following these events, three more immediate observations are made.

1. Files that are typically used to store data (e.g. .pdf, .jpeg, .txt) are encrypted and have the extension "`.WNRY`" appended to them.
2. On the network, a massive number of ARP packets[1] originate from the compromised host.
3. After some time, the desktop wallpaper is replaced with a note saying "Ooops, your important files are encrypted". The @WanaDecryptor@.exe pops up every minute or so, demanding a ransom payment of 300 USD in Bitcoin to the given wallet.

[1] In an actual, non-laboratory environment, these packets would likely be SMB connections as opposed to ARP requests. This is because one of the network interfaces on the Windows host is assigned a link-local address, and the sample attempts to find other hosts with link-local addresses, but there are none on the network.

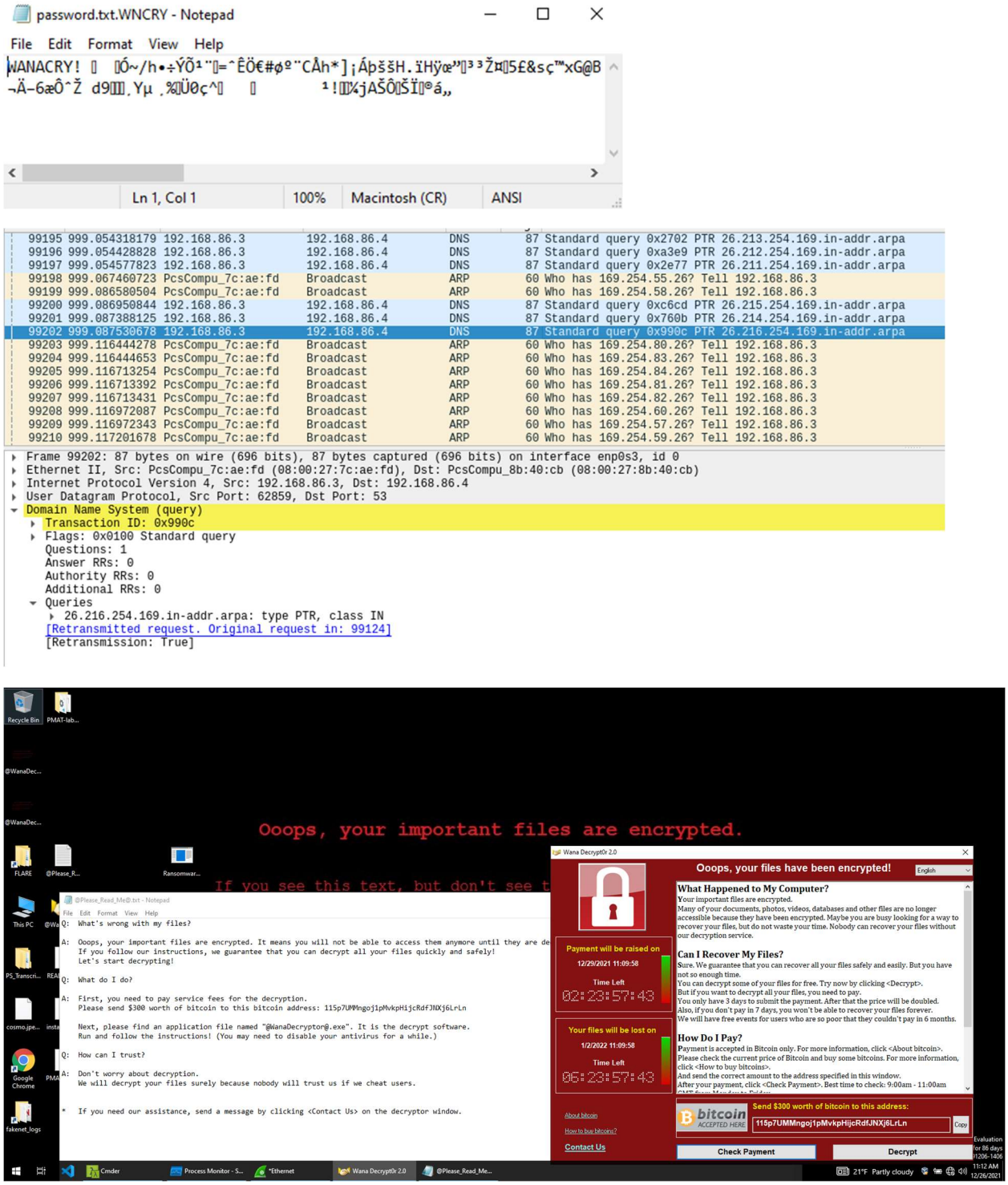


Fig 4.6: Various screenshots showing the indicators: (top to bottom) an encrypted text file, packet capture with large volume of ARP requests, compromised host desktop

Aside from this, persistence was found in the Services[2].

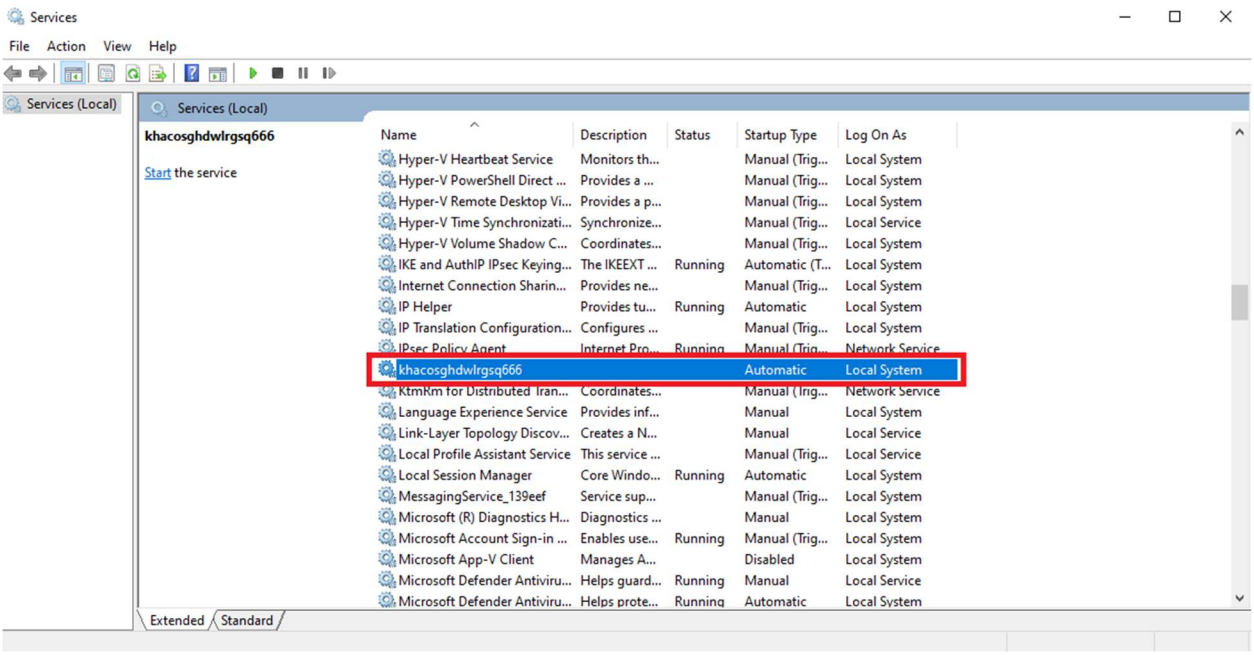


Fig 4.7: Screenshot from the Windows Services showing the creation of a new service.

To prevent victims from restoring their compromised machines to a previous state, the shadow copies used by the Volume Shadow Copy Service (VSS) are deleted.

```
cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet
\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
wmic  shadowcopy delete
taskse.exe C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe"
cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "khacosghdwlrgsq666" /t REG_SZ /d "\"C:\ProgramData\khacosghdwlrgsq666\tasksche.exe\"" /f
\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
reg  add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "khacosghdwlrgsq666" /t REG_SZ /d "\"C:\ProgramData\khacosghdwlrgsq666\tasksche.exe\"" /f
taskdl.exe
taskdl.exe
taskse.exe C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe"
taskse.exe C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrgsq666\@WanaDecryptor@.exe"
```

```
FLARE Fri 12/31/2021 17:23:18.86
C:\Users\sreisz>vssadmin list shadows
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
```

Fig 4.8: Evidence of the wiped shadow copies.

[2] Although the sample attempted to modify keys in the registry to gain additional persistence, these changes were not actually seen in the Registry. Further investigation may be necessary.

Interaction with Decryption Program

The three links at the bottom left of the “Wana Decrypt0r 2.0” program redirect to google searches about each topic. This leaves the “Check Payment” and “Decrypt” functions.

Clicking the “Check Payment” button leads us to discover that the tcp/9050 port is open. It seems to function as a server that connects to remote addresses over HTTPS. When the button is clicked, a connection is made to it, and an error message is sent back.

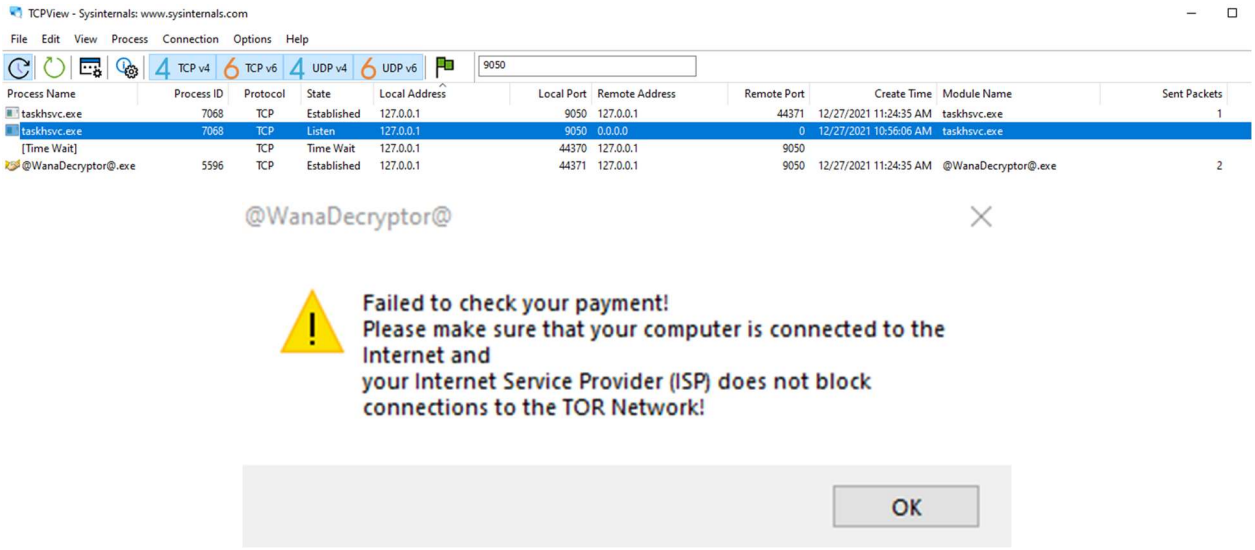
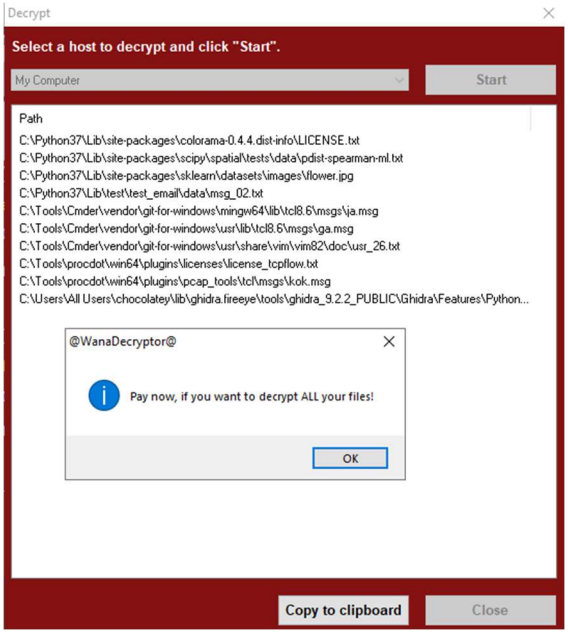


Fig 4.9: Evidence of the wiped shadow copies



The decryption program also gives victims a small “demo” of the decryption capability, likely to foster trust in the threat actors operating the malware. Clicking the “Decrypt” button shows a popup of the files that were decrypted, and then saying that payment was required to decrypt all files.

The full details of the encryption and decryption process have not been fully explored by this report at this time.

Fig 4.10: Decryption program.

Advanced Static Analysis

Having had a better understanding of the functionality of the WannaCry ransomware, the Advanced Analysis phases had the following goals:

1. Understand the “kill switch” mechanism.
2. Find more information on how the ransomware propagates.
3. Find any additional functionality that was not previously observed.

Screenshots of larger disassembly segments can be found in the appendices.

The “kill switch” mechanism can be found in the disassembly of the main() function (Appendix B, Fig 8.2). The execution can be described as such:

1. The previously noted URL is moved into the ESI register.
2. Parameters are pushed onto the stack to call InternetOpenA, and the result of the call is moved into EAX and EDI.
3. The JNE instruction is called. If the call was unsuccessful, continue with the flow of the program. Otherwise, if the call was successful, exit the program.

The propagation mechanism seems to be within an “mssecservice2.0” service that is separate from the previously discussed service for persistence. Looking at the disassembly for more information about this proved to be inconclusive.

```
0x0040808d      ret 8
161: fcn.00408090 ();
; var int32_t var_4h_3 @ esp+0xc
; var int32_t var_4h_2 @ esp+0x10
; var int32_t var_ch_2 @ esp+0x14
; var int32_t var_10h_2 @ esp+0x18
; var int32_t var_14h_2 @ esp+0x1c
; var char *lpServiceStartTable @ esp+0x20
; var int32_t var_ch @ esp+0x24
; var int32_t var_10h @ esp+0x28
; var int32_t var_14h @ esp+0x2c
0x00408090      sub esp, 0x10
0x00408093      push 0x104          ; 260
0x00408098      push 0x70f760
0x0040809d      push 0              ; HMODULE hModule
0x0040809f      call dword [GetModuleFileNameA] ; 0x40a06c ; DWORD GetModuleFileNameA(HMODULE hModule, LPSTR lpFilename, DWORD nSize)
0x004080a5      call dword [__p___argc] ; 0x40a12c
0x004080ab      cmp dword [eax], 2
0x004080ae      jge 0x4080b9
0x004080b0      call fcn.00407f20
0x004080b5      add esp, 0x10
0x004080b8      ret
0x004080b9      push edi
0x004080ba      push 0xf003f        ; '?'
0x004080bf      push 0
0x004080c1      push 0              ; LPCSTR lpMachineName
0x004080c3      call dword [OpenSCManagerA] ; 0x40a010 ; SC_HANDLE OpenSCManagerA(LPCSTR lpMachineName, LPCSTR lpDatabaseName, DWORD dwDesiredAccess)
0x004080c9      mov edi, eax
0x004080cb      test edi, edi
0x004080cd      je 0x408101
0x004080cf      push ebx
0x004080d0      push esi
0x004080d1      push 0xf01ff
0x004080d6      push str.mssecsvc2.0 ; 0x4312fc ; LPCSTR lpServiceName
```

Fig 5: Disassembled code that creates the “mssecservice2.0”

Since encryption was observed in the tasksche.exe binary, Advanced Static Analysis of the initial stage will not yield much more information.

Advanced Dynamic Analysis

A hypothesis was previously made that a successful callback to the URL (hxxp://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com) would cause the sample to stop execution. This can be tested by putting the ransomware through a debugger. After starting up an internet simulation, we can step through the executable until we hit the JNE instruction that would normally exit out of the routine.

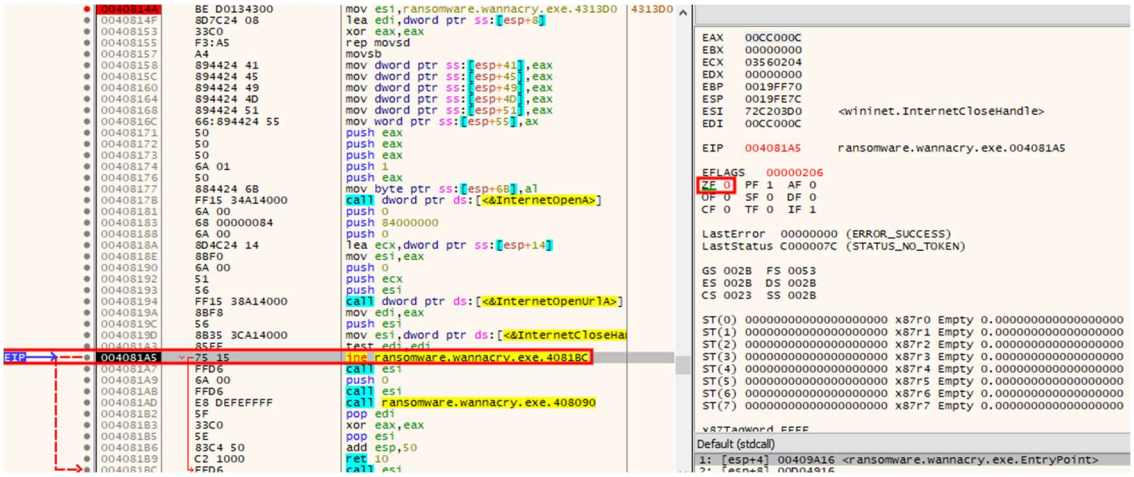


Fig 6.1: The debugger just before the jump should be followed

Using the debugger, we can set the zero flag to the Boolean value 1, such that we prevent the exit subroutine. Taking another step in the debugger proves our hypothesis, as the jump was not taken, and we can continue execution.

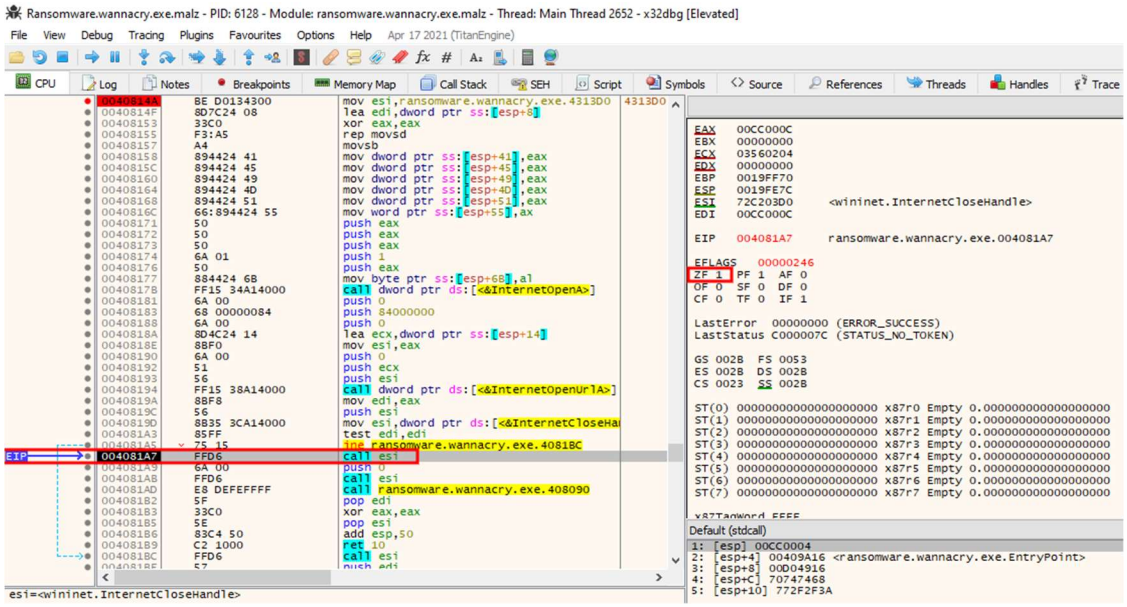
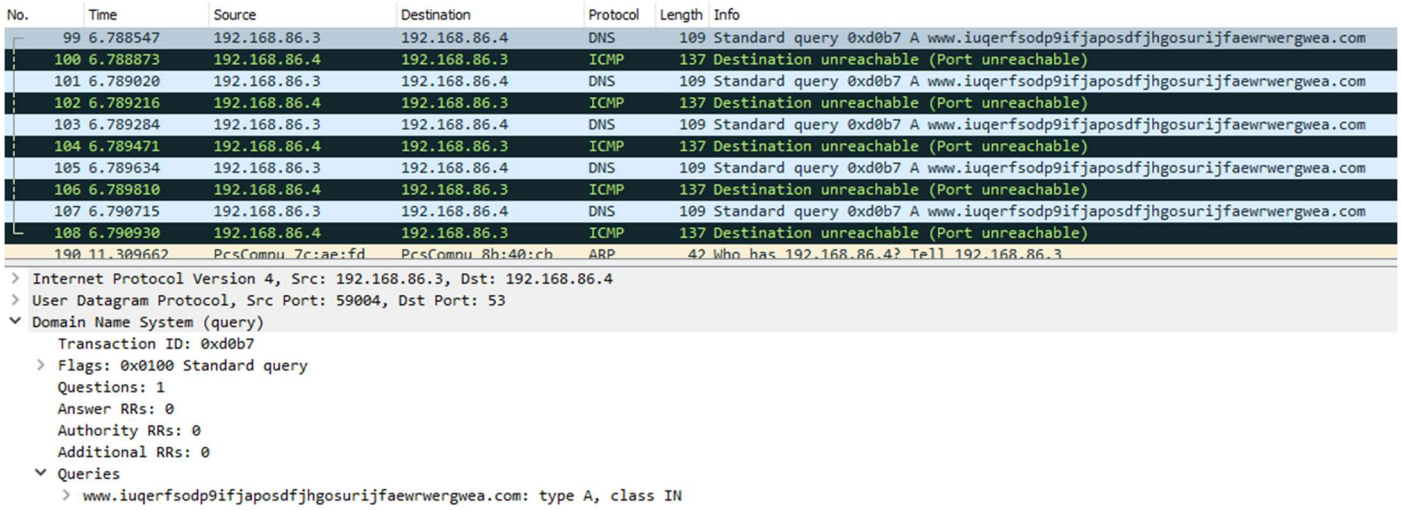


Fig 6.2: The result of switching the Zero Flag from 0 to 1

Indicators of Compromise

Network Indicators

1. Calls to the hxxp://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com URL. Note that this URL has since been registered in DNS records to act as a sinkhole.

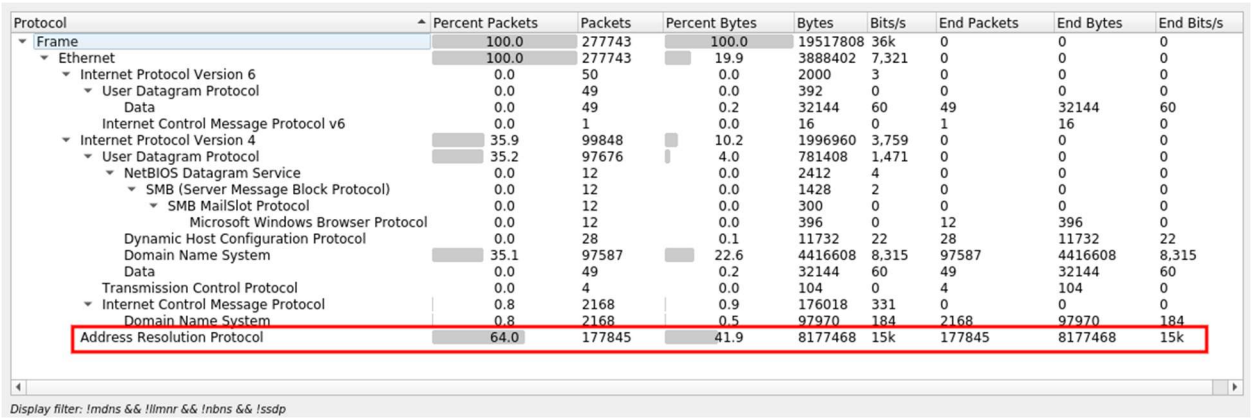


No.	Time	Source	Destination	Protocol	Length	Info
99	6.788547	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
100	6.788873	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
101	6.789020	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
102	6.789216	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
103	6.789284	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
104	6.789471	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
105	6.789634	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
106	6.789810	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
107	6.790715	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
108	6.790930	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
190	11.309662	PcsCompu_7c:ae:fd	PcsCompu_8b:40:cb	ARP	42	Who has 192.168.86.4? Tell 192.168.86.3

```
> Internet Protocol Version 4, Src: 192.168.86.3, Dst: 192.168.86.4
> User Datagram Protocol, Src Port: 59004, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0xd0b7
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com: type A, class IN
```

Fig 7.1: Wireshark Packet Capture of initial callback

2. Flood of ARP and/or SMB packets leaving the compromised host.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	277743	100.0	19517808	36k	0	0	0
Ethernet	100.0	277743	19.9	3888402	7,321	0	0	0
Internet Protocol Version 6	0.0	50	0.0	2000	3	0	0	0
User Datagram Protocol	0.0	49	0.0	392	0	0	0	0
Data	0.0	49	0.2	32144	60	49	32144	60
Internet Control Message Protocol v6	0.0	1	0.0	16	0	1	16	0
Internet Protocol Version 4	35.9	99848	10.2	1996960	3,759	0	0	0
User Datagram Protocol	35.2	97676	4.0	781408	1,471	0	0	0
NetBIOS Datagram Service	0.0	12	0.0	2412	4	0	0	0
SMB (Server Message Block Protocol)	0.0	12	0.0	1428	2	0	0	0
SMB MailSlot Protocol	0.0	12	0.0	300	0	0	0	0
Microsoft Windows Browser Protocol	0.0	12	0.0	396	0	12	396	0
Dynamic Host Configuration Protocol	0.0	28	0.1	11732	22	28	11732	22
Domain Name System	35.1	97587	22.6	4416608	8,315	97587	4416608	8,315
Data	0.0	49	0.2	32144	60	49	32144	60
Transmission Control Protocol	0.0	4	0.0	104	0	4	104	0
Internet Control Message Protocol	0.8	2168	0.9	176018	331	0	0	0
Domain Name System	0.8	2168	0.5	97970	184	2168	97970	184
Address Resolution Protocol	64.0	177845	41.9	8177468	15k	177845	8177468	15k

Display filter: !mdns && !llmnr && !nbns && !ssdp

Fig 7.2: Wireshark statistics output noting that 64% of protocols were ARP

3. A process, taskhsvc.exe, listening on port tcp/9050. Attempts to make connections with non-private IP addresses may also be occurring. Note that the sha256sum of this file matches the tor.exe file found in the staging directory, implying possible "dark web" communication.

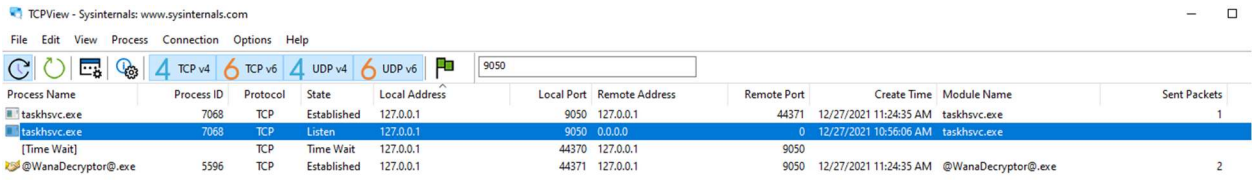


Fig 7.3: Screenshot from TCPView depicting taskhsvc.exe activity

Host-based Indicators

1. The directory shown Fig 2 is created, containing a similar array of files. Note that the name of the directory may be different from host to host, but it will be hidden.
2. The existence of a new service, with the same name as the hidden directory, will be created, which gives tasksche.exe persistence

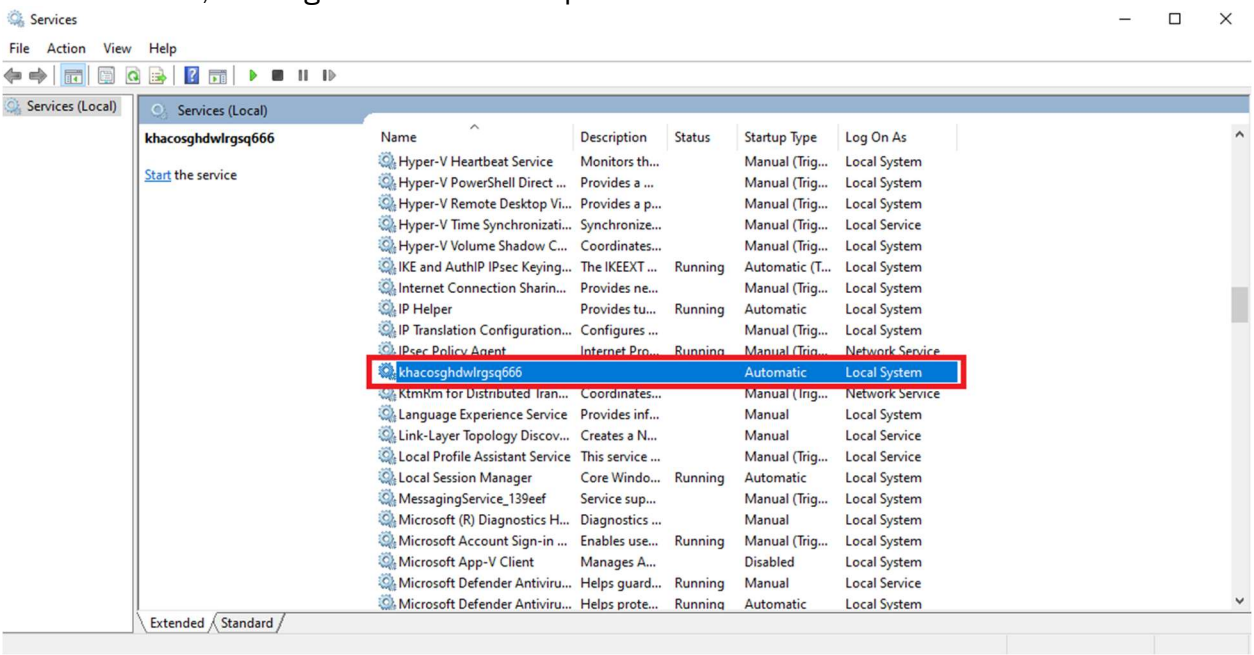


Fig 7.4: Screenshot of the newly created, malicious service

3. A visibly changed desktop, with the WanaDecrypt0r ransom note periodically appearing at the top of a user's view (Shown in Fig. 4.6).
4. Shadowed copies of the filesystem are wiped from the system (Shown in Fig. 4.9).

Rules & Signatures

A full set of YARA rules is included in Appendix A.

As the IOCs from the previous section have shown, this ransomware sample operates very overtly, and as such, has very apparent signatures. Not all were used in the rules developed, in an attempt to generalize the usage, however, they are usable nonetheless.

- The ransomware makes it obvious what its name is. This will show up in various strings throughout the binary, such as “WANACRY!”, “wnry”, “WanaCrypt0r”, etc.
- The presence of Bitcoin wallet addresses in the strings, while not necessarily an indicator of WannaCry, is reason to be suspicious.
    - 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
    - 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
    - 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
- Callbacks to URLs like the one discussed in this report gives high confidence that a sample may be WannaCry. Since the original attacks in 2017, many of these URLs have since been sinkholed, but source code could easily be modified.
- High volumes of SMB traffic and/or ARP requests originating from one or more hosts
- Anything to do with the Indicators of Compromise referenced throughout this document

Appendices

A. Yara Rules

Full repository located at: http://github.com/An00bRektn/malware-analysis-reports

```
rule Ransomware_Wannacry {
    meta:
        last_updated = "2021-12-30"
        author = "An00bRektn"
        description = "A simple yara rule to detect WannaCry"

    strings:
        $PE_magic_byte = "MZ"
        $x1 = "__TREEID__PLACEHOLDER__"
        $x2 = "__USERID__PLACEHOLDER__"
        $x3 = "wnry"

        // Catch hardcoded SMB syntax
        $s1 = /\\\\(\d{1,3}\.){3}\d{1,3}\\IPC\$/ nocase fullword wide
        $s2 = "\\\\%s\\IPC$" fullword ascii

        // Note that these two probably change between variations
        $s3 = "WANACRY!" fullword ascii
        $s4 = "WanaCrypt0r" fullword ascii
        $s5 = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii

    condition:
        $PE_magic_byte at 0 and
        ($x1 and $x2) and ($s1 or $s2) and ($s3 or $s4) or $x3 or $s5
}
```

Fig 8.1: General YARA rule to identify WannaCry

B. Disassembled Code Snippets

Fig 8.2: Disassembled main function in Cutter.